

隨堂練習卷

建議作答 40 分鐘 · 總分 100

姓名： _____ 學號： _____ 班級： _____
得分： _____

作答說明：紙筆測驗，請勿使用電腦。程式題寫出完整可編譯程式碼，注意大小寫、分號與大括號。

§ 1 一、選擇題 (每題 6 分，共 30 分)

- 每個遞迴方法都必須具備哪兩個部位？ (A) 迴圈與計數器 (B) 基底條件與遞迴步驟 (C) 參數與回傳值 (D) try 與 catch
- 關於「基底條件 (base case)」，下列何者正確？ (A) 它一定要呼叫自己 (B) 它是最簡單、可直接回答的情況，遞迴的終止點 (C) 一個遞迴方法只能有一個基底條件 (D) 它讓問題變得更大
- 漏寫基底條件、或遞迴步驟不收斂，最可能發生什麼？ (A) 編譯錯誤 (B) 回傳 0 (C) 無限遞迴、丟 `StackOverflowError` (D) 自動改用迴圈
- 關於遞迴在呼叫堆疊上的行為，下列何者正確？ (A) 遞迴有特殊機制、不用堆疊 (B) 每次遞迴 push 一個活動記錄，到基底後依序 pop (LIFO) (C) 所有遞迴框共用同一份參數 (D) 先 push 的框最先 pop
- 關於遞迴與迭代，下列何者正確？ (A) 遞迴一定比迭代快 (B) 兩者不可互換 (C) 迭代不需額外堆疊框、遞迴每次呼叫多一個框 (D) 迭代一定會 `StackOverflowError`

答：1. _ 2. _ 3. _ 4. _ 5. _____

§ 2 二、遞迴展開與堆疊 (共 16 分)

- 下列遞迴呼叫 `factorial(4)`。寫出它的展開與回溯過程，並寫出最後回傳值 (8 分)：

```
public static long factorial(int n) {  
    if (n <= 1) return 1;  
    else return n * factorial(n - 1);  
}
```

展開到基底條件：`factorial(1) = _`；回溯往回乘：`factorial(2) = _`、`factorial(3) = _`、`factorial(4) = _____`。

1. 預測下列程式輸出，並說明 `fib(4)` 共呼叫了 `fib` 幾次 (8 分)：

```
public static long fib(int n) {
    if (n == 0 || n == 1) return n;
    else return fib(n - 1) + fib(n - 2);
}
public static void main(String[] args) {
    System.out.println(fib(4));
}
```


§ 3 三、改錯 (每題 6 分，共 18 分)

各有一個錯誤，指出並改正。

1. 想用遞迴算階乘，但漏了基底條件：`static long fact(int n){ return n * fact(n - 1); }`
2. 想用遞迴求和 1..n，遞迴步驟沒收斂：`static int sum(int n){ if (n <= 0) return 0; else return n + sum(n); }`
3. 想用遞迴算 gcd (輾轉相除)，但基底條件與遞迴引數寫反：`static int gcd(int a, int b) { if (a == 0) return b; else return gcd(a % b, b); }`

